

SPARK Lift

method map(row)

```
list <- []  
movie_dict <- {}  
movie_list <- split the row  
sort movie_list  
for all movie_i in movie_list do  
    for all movie_j > movie_i in movie_list do  
        if movie_j in movie_dict do  
            movie_dict[movie_j] <- movie_dict[movie_j] + 1  
        else  
            movie_dict[movie_j] <- 1  
    list.append((movie_i, movie_dict))  
    movie_dict <- {}  
return list
```

method reduce(data, new_data)

```
movie_i, movie_dict <- data  
movie_j, new_movie_dict <- new_data  
new_movie_count <- 0  
movie_count <- 0  
for all movie in new_movie_dict do  
    new_movie_count <- new_movie_count +  
    new_movie_dict[movie]
```

for all movie in movie_dict **do**

 movie_count <- movie_count + movie_dict[movie]

movie_i <- movie_i + "_" + movie_count + "," + movie_j + "_" +
new_movie_count

return (movie_i, {})

method create_movie_dict()

movie_dict <- {}

movie_strings <- split reduced_RDD

for all movie_data in movie_strings **do**

 movie_id, movie_count <- split movie_data

if movie_id in movie_dict **do**

 movie_dict[movie_id] <- movie_dict[movie_id] +
 movie_count

else

 movie_dict[movie_id] <- movie_count

return movie_dict

method filter(row)

movie_i, movie_dict <- data

list <- []

movie_name_1 <- global_movie_names[movie_i]

for all movie in movie_dict **do**

 lift <- (movie_dict[movie] * total_users) /
 (global_movie_dict[movie] * global_movie_dict[movie_i])

if lift > threshold:

```
        movie_name_2 <- global_movie_names[movie]
        list.append((movie_name_1, movie_name_2, lift))

    return list
```

method load_movie_names()

read movies.csv file and store the movie names into a global dictionary where movie_id is key and movie_name is the value

method main()

```
    read text file as RDD
    RDD.map(split row into user_id, movie_id, rating)
    RDD.filter(rows with rating >= 4.0)
    RDD.map(row <- (user_id, movie_id))
    RDD.map(for each user_id concatenate all movie_id's)
    RDD.map(row <- concatenated movie_id's)
    total_users <- RDD.count()
    RDD.map(map)
    reduced_RDD <- RDD.reduce(merge the stripes for each movie)
    global_movie_names <- load_movie_names()
    global_movie_dict <- create_movie_dict()
    RDD.map(filter)
    save RDD as text file
```